Approach for Coupled Heat Transfer / Heat Flux Calculations

Stephan Bock
MTU Aero Engines GmbH
Dachauer Straße 665
D-80995 München
Germany
Fax.: +49-1489-97878
e-mail: stephan.bock@muc.mtu.de

Coupled CFD/FE heat transfer / heat flux calculations for material temperature prediction with independent grids expose the need to exchange boundary condition information between the fluid and the solid domain. The author introduces a method of storing the boundary condition information together with the geometry information, where boundary conditions are represented by a set of geometric entities. This method of "virtual boundary conditions" simplifies some aspects of the coupled calculations by introducing a clear grid interface and enables a master model approach from the geometry generation (CAD) down to the analytical disciplines (CAE). The detachment of the boundary condition information from the computational grid also exposes advantages when considering different degrees of geometric complexity (2D/3D).

Introduction

Modern jet engines with their highly loaded components expose a need to use advanced three-dimensional analytical tools to predict correct life times during design. The progress in computer aided design (CAD) and computer aided engineering (CAE) has enabled the use of three dimensional models and computations for almost every part of a jet engine. Coupled heat transfer / heat flux calculations represent a tool for very accurate prediction of temperatures in structural elements. They calculate the amount of heat taken into or from the structural element by the surrounding gas flows by modeling the gas flows themselves and the heat flux inside of the structure. Nevertheless, these calculations require extensive exchange of information between the different computational domains, which makes this information exchange a key problem to overcome.
Even though it is possible today to automatically create grids from CAD models, it is still very uncommon to also represent boundary conditions or calculation results inside the CAD model. Therefore, the idea of a master model that drives the entire design up to manufacturing is currently excluding the analytical disciplines. In this paper, the author introduces a method of storing and exchanging boundary conditions for calculations together with the CAD geometry in order to simplify the design process and coupled calculations in order to come a step closer to generating a complete master model.

The concept of "virtual" boundary condition objects

A typical design process starts from simple 1D and 2D geometries and evolves through 2.5D until the complete 3D model is designed. The analytical disciplines accompany this process by increasingly complex calculations where the boundary conditions stay always attached to the used grids and not to the underlying geometric model. The idea of attaching the boundary conditions to the geometry rather than the grid makes use of the fact that almost every software describes curves and surfaces in a parametric fashion - mostly B-splines. A boundary condition can now be regarded as a geometric element itself, which is called a "virtual" boundary condition that is associated with an element of the geometric model. Picture 1 depicts how a temperature boundary condition T for a model curve is stored as a "virtual" curve with identical parameterization in the plane y=0. For easy access, the x-coordinate of the virtual curve is chosen to be identical with the u-parameter and the z-coordinate corresponds to the value of the boundary condition.

Paper presented at the RTO AVT Symposium on "Advanced Flow Management: Part B – Heat Transfer and Cooling in Propulsion and Power Systems", held in Loen, Norway, 7-11 May 2001, and published in RTO-MP-069(I).

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **00 MAR 2003**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Approach for Coupled Heat Transfer/Heat Flux Calculations**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **NATO Research and Technology Organisation BP 25, 7 Rue Ancelle, F-92201 Neuilly-Sue-Seine Cedex, France**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
Also see ADM001490, presented at RTO Applied Vehicle Technology Panel (AVT)Symposium held in Leon, Norway on 7-11 May 2001, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**10**	

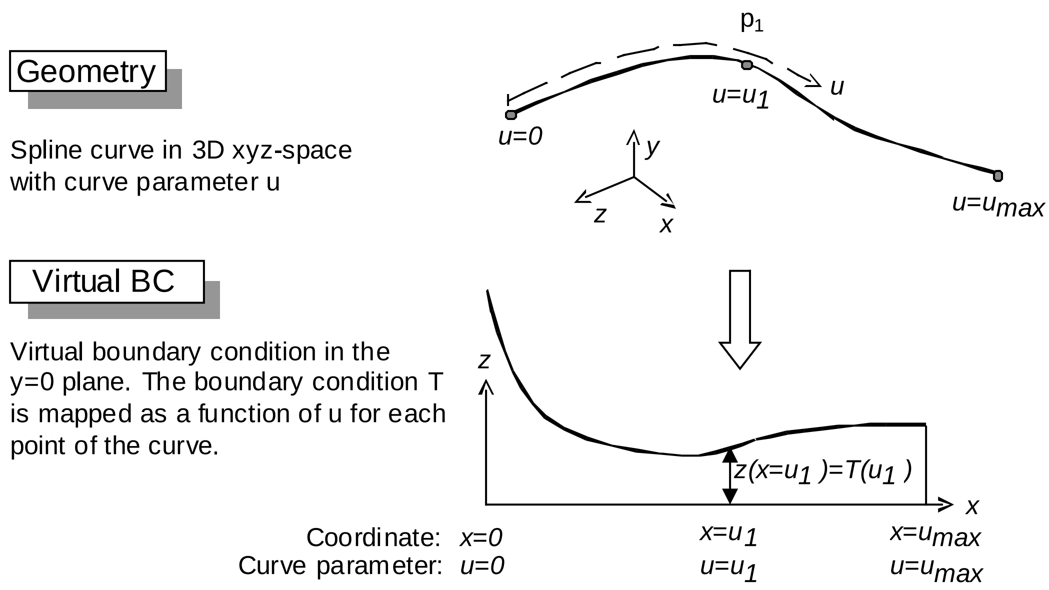

Figure 1:Schematic representation of a geometric entity and a corresponding virtual boundary condition element in 2D space.

For each possible point p_1 on the geometric curve, a boundary condition value can now be assigned by requesting the curve parameter u_1 of that point and reading the z-coordinate of the virtual curve at $x=u_1$. With the storage of boundary condition information in this manner, a new curve is necessary for each boundary condition (e.g. temperature, pressure etc.) that should be associated with the original curve. This leads to a structure, in which each curve uses a set of corresponding virtual boundary condition curves to represent the entire boundary condition information. The same schematic can also be applied to surfaces that have a (u,v) parameterization. For each surface in the model, a set of virtual surfaces can be created to store the boundary condition information. The u-parameter of these virtual surfaces corresponds to the x-coordinate and the v-parameter to the y-coordinate, while the z-coordinate holds the value of the boundary condition respectively, as can be seen in figure 2.

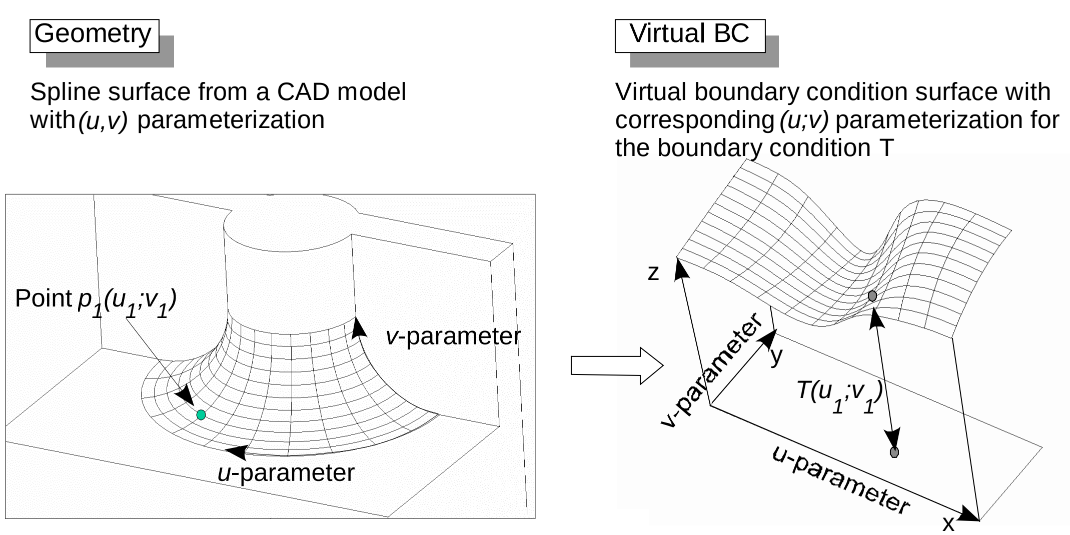

Figure 2: Representation of a surface and a corresponding virtual boundary condition element in three dimensional space.

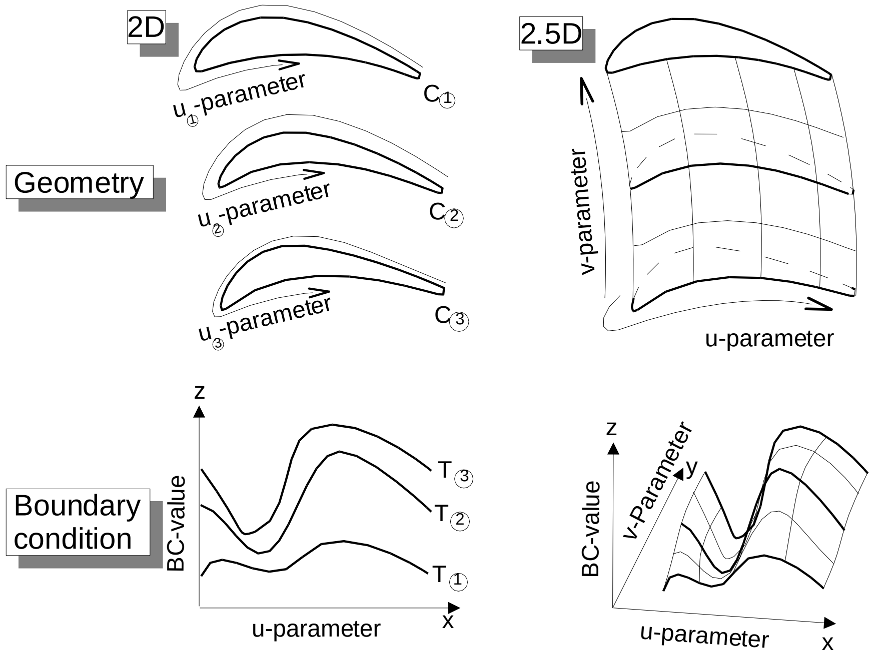

Figure 3: Creation of 2.5 dimensional geometric objects from 2D curves via lofting. The virtual boundary condition objects are lofted as well to create a corresponding 2.5D boundary condition.

This way of storing the boundary condition information has advantages in the progress of a typical design process. Since 2 dimensional models and analysis characterize the early stages of a new design, this information can later be used to switch to higher dimensional models as can be seen in figure 3.

In this example of an airfoil design, the shape of the airfoil is represented by three curves C_1-C_3 with corresponding virtual boundary condition curves T_1-T_3. The generation of 2.5 dimensional geometry is carried out by lofting a surface through the curves C_1-C_3. The boundary condition for this new surface is created accordingly, by lofting the virtual boundary condition curves T_1-T_3 to create a virtual boundary condition surface. The boundary condition information can then be used as a starting point for the 2.5D analysis.

Accessing the information of virtual boundary conditions

Once the boundary condition information is stored in the "virtual" form, that means as geometric entities, it can easily be accessed by any program simply by dropping the (x,y,z) coordinates of each grid node onto the

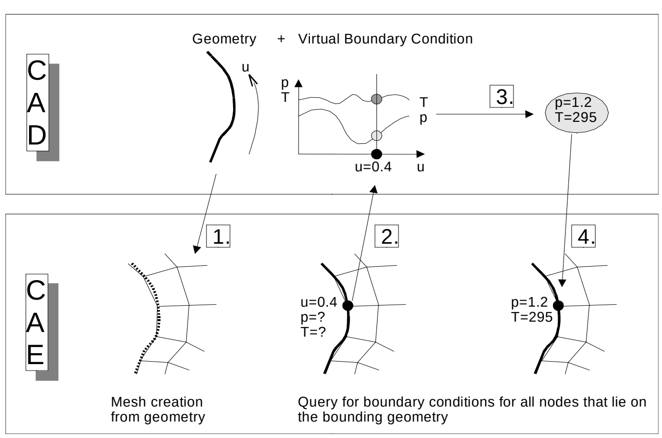

Figure 4: Extraction of boundary condition values from the virtual boundary conditions. A point on a mesh that was created from an existing geometry (1.) is dropped onto the geometry (2.) and boundary condition values are extracted from the virtual objects (3.). These values are then imposed onto the mesh (4.).

surface and obtaining the (u,v) parameters for this location. With these parameters, the z-coordinate value of the "virtual" boundary condition surface can be read out and returned to the analytical program as boundary condition value at this node as visualized in figure 4. The close interaction between CAD, where the geometry and the boundary conditions are stored, as well as CAE becomes obvious with this approach.
The computational power required for the determination of the u and v parameters as well as the extraction of the boundary condition at any given point is small because highly optimized subroutines exist for these tasks in virtually every standard software package that uses spline surfaces.

Application for coupled calculations

Coupled calculations usually incorporate different solvers for the domains to be investigated. A common application is the coupling of CFD solvers for fluids and FE solvers for solid structures. Each solver uses a grid to resolve its computational domain and the exchange of information has to be managed by interpolation between grids. Figure 5 depicts that situation, showing the different grids and their common border curve where the grids need to exchange information. The use of virtual boundary condition objects does not eliminate the need for interpolation, but it significantly reduces the complexity of the problem by creating a clear interface for information exchange. Nevertheless, for coupled calculations the situation is more complex since the virtual boundary condition objects have to be recreated from the results of each single computation. The results are typically attached to the computational grid and have no connection to the underlying geometry. Therefore, this information has to be stored in an independent file from where it can be accessed to recreate a boundary condition object after the calculation.

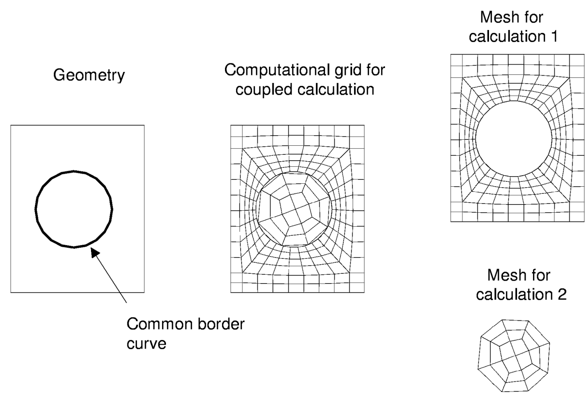

Figure 5: Schematic process of coupled heat transfer/heat flux calculations using independent meshes for the fluid and solid domain. The virtual boundary condition elements are used to exchange information at the common border.

The general process used for coupled calculations is explained in figure 6. The first step (1) creates the grid for one computational domain (e.g. the solid region) from the geometry where the (u,v) parameter information for each node is stored externally (2). This information together with the virtual boundary conditions leads to the starting grid with imposed boundary condition values at each node (3) from where the solver calculates a first solution (4)-(5). These results are then re-transformed into virtual boundary condition objects, again with the use of the (u;v) parameter data (6). Then, the whole process is repeated for the other computational domain (e.g. the fluid region) until a converged solution is obtained. It is not necessary, that all boundary conditions are rebuilt after each run, sometimes it is even impossible because the boundary condition is input and not a result. For the CFD/FE calculation, for example, the boundary conditions for the fluid domain can be constant temperature from which the wall heat flux is computed and added to the list of boundary condition objects. The solid domain calculation uses this boundary condition to calculate a material temperature, which then modifies the existing temperature boundary condition for the fluid domain. By repeating this scheme, a

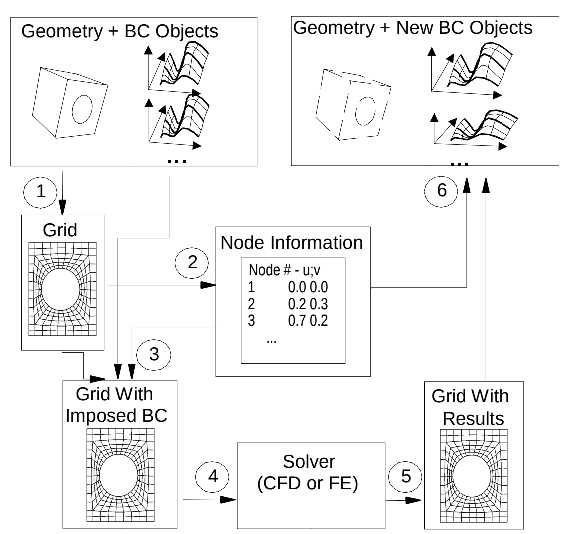

Figure 6: Complete process of probing for the use and recreation of boundary conditions

coupled heat flux - heat transfer calculation can be performed. The obvious difficulty with this approach is the recreation of the virtual boundary condition object from the discrete (node or element centered) values of the calculation results. This represents a best-fit problem for curves or surfaces from a cloud of points. With block structured meshes, the best fit of a surface is still fairly simple because the nodes are ordered. For the general case of unstructured meshes, the interpolation of a surface through the resulting points is more complex as can be seen in figure7.

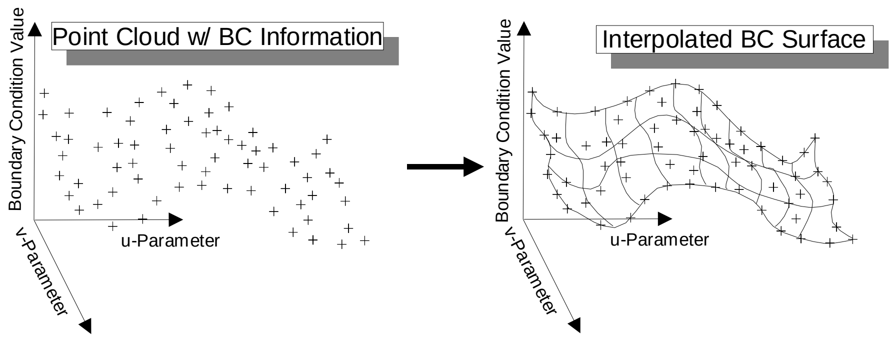

Figure 7: Interpolation of a new virtual boundary condition surface from result point data

Surface fitting from calculated results

There are many good algorithms for surface fitting through irregular data points and a good overview is given in [8]. For the sake of simplicity in programming and its good performance with large numbers of fairly regularly distributed points, a cell oriented interpolation method similar to the modified Shepard's algorithm developed by RENKA [6] with a regular interpolation grid was chosen. The algorithm creates a rectangular grid of interpolation points in the point cloud through which the surface can then be fitted easily as show in figure 8.

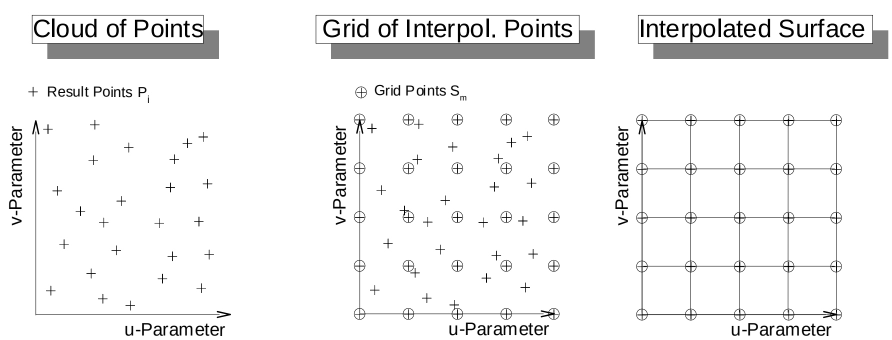

Figure 8: Creation of a regular grid of interpolation points used to create a surface from a point cloud.

The first step to create an interpolation point is to find its closest neighbors and then use the distance-weighted interpolation with these points to calculate the needed values R_{iS}

$$R_{iS} = \frac{\sum_{m=1}^{M} \frac{R_{im}}{d_m^2}}{\sum_{m=1}^{M} \frac{1}{d_m^2}}$$

with:

- R: Value of boundary condition
- i: Index of boundary condition (in the case of multiple boundary conditions, e.g. pressure and temperature)
- S: Interpolation point with parameters (u_S,v_S)
- m: Index denominating all neighboring points for interpolation (up to M)
- d: Distance between point P_m and interpolation point S

With the cell oriented algorithm, the number of comparisons needed to find the closest neighbors decreases dramatically, because only points in concentrical rings of neighboring cells have to be considered as shown in figure 9. Depending on the number M of data points considered for the interpolation of each grid point and the cell size, a numerical smoothing can be imposed to the resulting surfaces, leading to a stabilization of the coupled computing algorithm described above. For the test calculations presented in this paper, eight surrounding points for each interpolation point produced good results. In most cases, the amount of computing power is reduced significantly when compared to the direct interpolation from one grid to another, because only relatively few interpolation points have to be determined to span a surface over the resulting data.

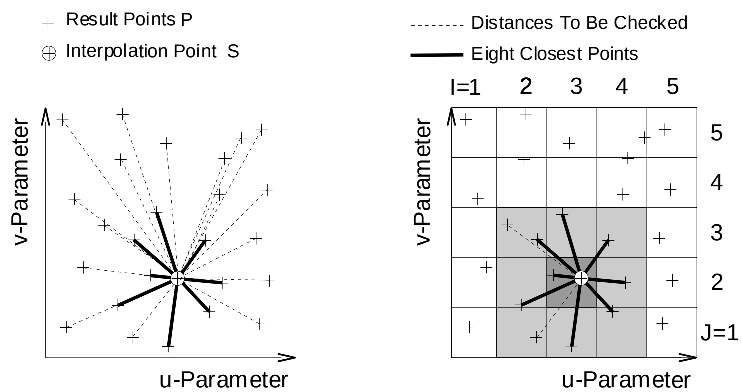

Figure 9: A cell oriented method reduces the amount of comparisons needed to determine the next neighbors by searching in cells around the interpolation point.

A coupled heat flux / heat transfer calculation with virtual boundary conditions

As described above, the coupling of heat flux / heat transfer calculations is used to demonstrate the method of “virtual” boundary conditions because of the need to exchange information over boundary conditions between the fluid and solid domain. The geometry used to perform the calculations was experimentally investigated by BERG [1], ELFERT [3] and HEIN[10]. It consists of a tube with a length of 20 hydraulic diameters and a smoothly rounded inlet area as shown in figure 10. The setup was designed to investigate the influence of rotation to local Sherwood/Nusselt number distributions. It was later used to calibrate several CFD calculations ([5] [2] [7]). In extension of the experiment, a computation makes it possible to investigate the influence of wall thickness and external heat transfer.

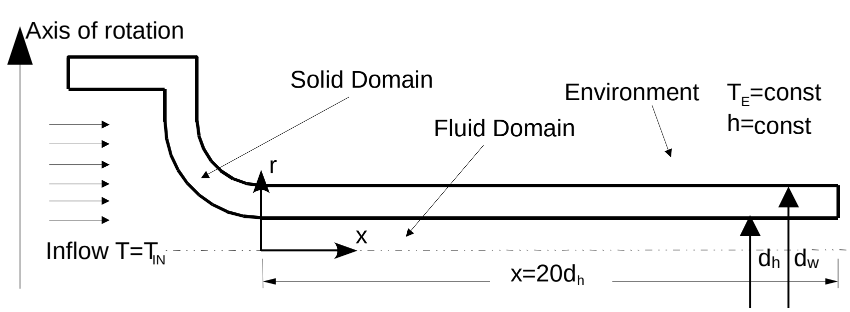

Figure 10: Schematic layout of the experimental setup used by Berg [1] and Elfert [3].

Therefore, the wall is modeled as solid domain for the heat flux calculations and the inside of the tube is modeled as the fluid domain. Figure 11 shows a coarse version of the grids used for the calculations with a modified logarithmic wall function for heat transfer on the inside of the pipe and constant heat transfer from the environment of the tube.
The coupling was established using the method introduced by HESELHAUS [9] in order to guarantee a stable algorithm. This method calculates artificial heat transfer coefficients with local temperature differences at each node of the wall and uses these as input for the heat flux calculation. After the finish of the heat flux calculation, the resulting solid material temperature is used again as boundary condition for the fluid flow. The entire setup consists of two independent grids derived from several surfaces, each equipped with a set of "virtual" boundary conditions to store and exchange the necessary data between these grids.

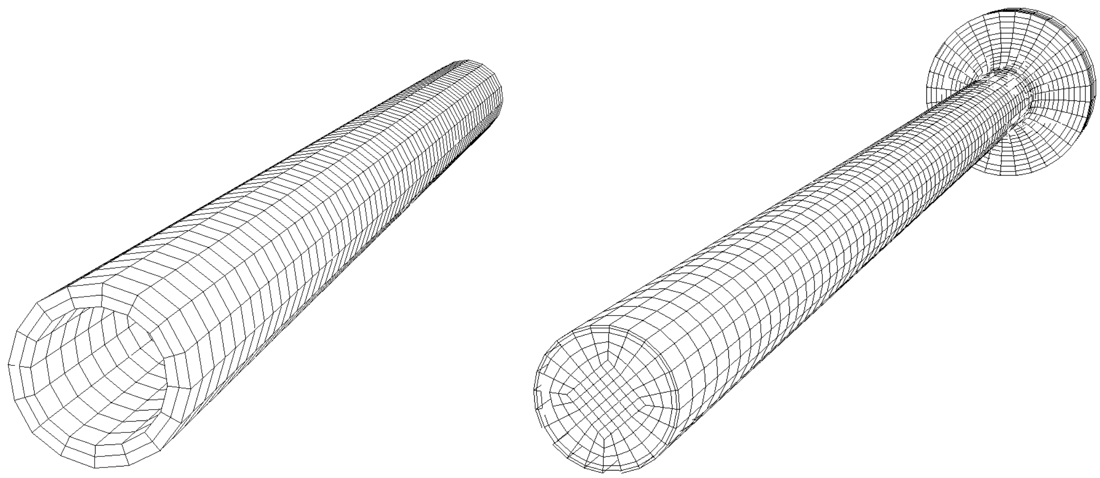

Figure 11: Computational domains used for coupled calculation (solid domain on the left and fluid domain on the right)

The geometry was created using the program `ICAD` from Knowledge Technologies International (KTI) which provides a geometry kernel in combination with an object oriented programming language. The concept of "virtual" boundary conditions was employed by creation of a new surface class incorporating a list of supplementary surfaces to represent the boundary conditions as attribute. Methods to query boundary condition values and to recreate boundary condition surfaces from a cloud of points were employed and linked to that surface class. The CFD computations were carried out using `CFX TASCflow V2.10` with a standard k-ε turbulence model as well as modified logarithmic wall functions [4]. The known shortcomings of this approach for the prediction of accurate heat transfer were accepted because the focus of the calculations was the proof of concept for the boundary condition representation. The heat flux calculations for the solid domain were carried out using the `MSC Qtran` software.

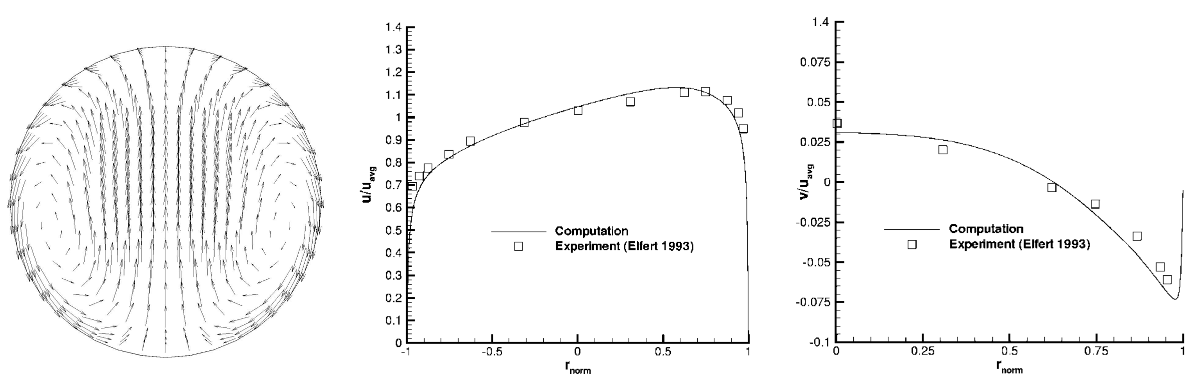

Figure 12: Velocity vectors showing the secondary flow field at the end of the pipe (left) and comparison with the experiments of Elfert [3].

Figure 12 shows the visualization of the secondary flow field at the end of the tube at $x/d_h=20$ for a case with Re=45000 and Ro=0.044 as well as a comparison of the main flow velocity u in the plane of rotation (symmetry) and the secondary flow v perpendicular to the plane of rotation. The comparison exposes a good match of the flow field to the experiments conducted by ELFERT and HEIN.

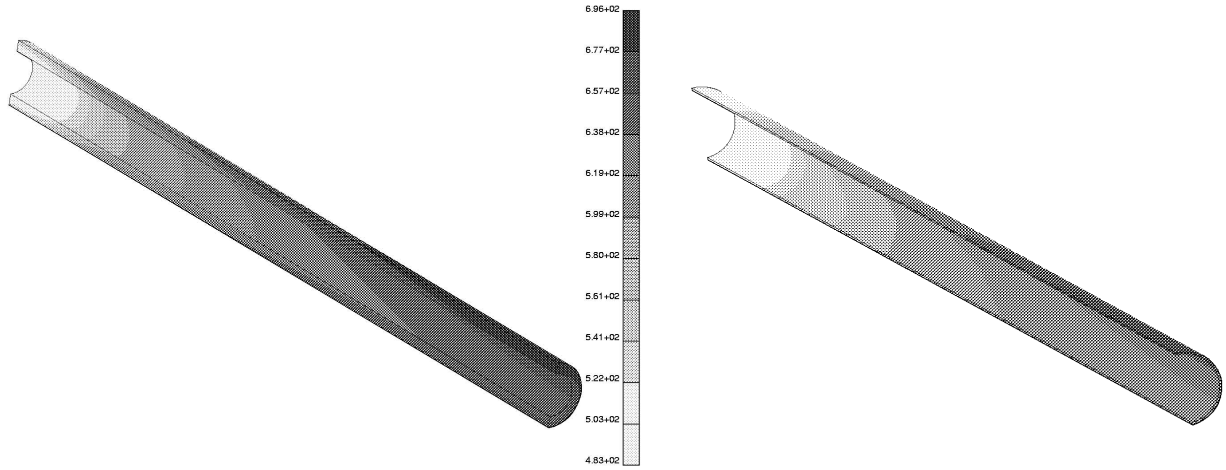

Figure 13: Material temperatures in the pipe wall for different wall thickness

With the coupling algorithm by HESELHAUS, a material temperature in the wall of the pipe was calculated with a constant environment temperature T_E=800K and constant heat transfer coefficient h=400 on the outside of the pipe. The fluid inlet temperature T_{IN} was 290K and the fluid heated up to about 400K at the outlet. The material temperature for a wall thickness of $d_w/d_h=1.5$ and $d_w/d_h=1.0625$ is shown in figure 13.
With this information, the Nusselt number distribution on the inside of the pipe can be calculated. This distribution shows some discrepancies with the results obtained by BERG which are easily explainable by the use of the simple k-ε turbulence model and wall functions. Nevertheless, the general trend as displayed in figure 14 is in good accordance with the experiments published in [1].

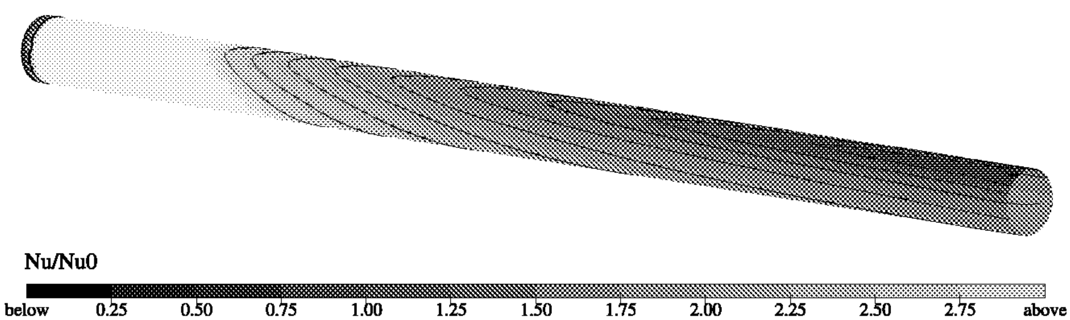

Figure 14: Nusselt number distribution on the inside of the pipe wall.

Conclusions

The method of virtual boundary condition objects is implemented and works well for coupled heat transfer / heat flux computations. With the geometric representation of boundary conditions, a close connection between CAD and CAE is possible whereas the "virtual" objects act as interfaces. The boundary condition is stored in one central place together with the geometry and can be accessed by any grid generator to impose values onto the nodes generated. Therefore, the boundary condition becomes independent of the used grid type which enables easy comparison of results obtained with different meshes. The interpolation time can be reduced because only relatively few points are needed to describe the boundary condition surface. The

boundary condition follows the geometry generation when the step from 2D to 3D is carried out. The interpolation from resulting nodes to a smooth surface works well and can be carried out by effective interpolation schemes. Together with modern CAD systems and grid generators, the representation of boundary conditions as “virtual” surface objects extends the master model philosophy into the analytical disciplines.

Acknowledgements

The research for this publication was conducted in close cooperation between the “Deutsches Zentrum für Luft- und Raumfahrt DLR ” and MTU as part of the framework “DASA/DLR Nachwuchsinitiative”.

Bibliography

[1] H. P. Berg. Experimentelle Bestimmung des örtlichen inneren Wärmeübergangs von Turbinenleit- und -laufschaufeln mit Hilfe der Analogie zwischen Wärme- und Stoffübergang. Dissertation, Technische Universität Darmstadt, 1991

[2] M. de Martino. Numerische Berechnungen des Stoffübergangs in einem idealisierten Kühlkanal einer Turbinenschaufel. Diplomarbeit, Technische Universität Darmstadt, 1999.

[3] M. Elfert. Wärmeübergang in rotierenden Kühlkanälen von Gasturbinenschaufeln, Deutsche Forschungsanstalt für Luft - und Raumfahrt (DLR), Januar 1993. Abschlußbericht BMFT Vorhaben 0326500E, AG-Turbo, Turbotherm, Bauteilkühlung, Vorhaben 2.1.4.2.

[4] H. Grotjans, F. R. Menter. Wall Functions for General Application CFD Codes. In Proceedings ECCOMAS 98, John Wiley & Sons, Ltd., 1998.

[5] C. Kühner. Entwurf von rotierenden Kühlkanälen in Gasturbinenschaufeln auf Basis numerischer Analyse der Strömungen und Druckverluste. Konstruktiver Entwurf, Deutsches Zentrum für Luft- und Raumfahrt (DLR) – Institut für Antriebstechnik, 1997.

[6] R. J. Renka. Multivariate Interpolation of Large Sets of Scattered Data. In ACM Transactions on Mathematical Software, volume 14, No. 2, pages 139-148, June 1988.

[7] C. Sivade. Numerische Untersuchung des Stoffübergangs in einem idealisierten Kühlkanal einer Turbinenschaufel. Diplomarbeit, Technische Universität Darmstadt, 2000.

[8] D. Watson. CONTOURING: A guide to the analysis and display of spatial data. Pergamon Press, 1992. ISBN 0-08-040286-0.

[9] A. Heselhaus. Ein hybrides Verfahren zur gekoppelten Berechnung von Heißgasströmung und Materialtemperaturen am Beispiel gekühlter Turbinenschaufeln, Dissertation, Ruhr-Universität Bochum, DLR-Forschungsbericht 1997-10, ISSN 0939-2963, 1997.

[10] O. Hein. Untersuchung der Strömung in einem radial gerichteten Kühlkanal eines Turbinenlaufrades, Dissertation, Ruhr-Universität Bochum, DLR-Forschungsbericht 1999-33, ISSN 1434-8454, 1999.

Paper Number: 34

Name of Discusser: J.J. Mc Guirk, Loughborough University, England

Question:
My question concerns the interpolation procedure for generating new virtual boundary conditions. Before the interpolation, the temperature filled in the point cloud will imply a certain total heat flux over a surface from (say) the fluid flow into the solid material.
How do you know that that same heat flux is represented by the interpolated surface?
If not then the heat flow is not concerned between fluid and solid regions, the interpolation procedure may create or distroy heat flux?

Answer:
The conservation is necessary in any kind of interpolation Scaling of the virtual boundary condition surface can be one way of adjustment to account for this.
A comparison of the calculated total heat flux and the generated boundary condition surface has to be carried out to determine the scaling factor.

Name of Discusser: De Luca L., Politecnico Milano, Italy

Question:
Impressive 3-D calculations were shown in the previous presentation. Your technique is a interesting mathematical idea.
What specific advantages do you expect from the already available routines:
Accuracy of the results? Stability of procedure executing time?

Answer:
1 Process of boundary condition implementation is simplified
2 Independent of used mesh, especially refinement and type
3 Gain in interpolation speed
4 Easy adjustments for changes in geometry
5 Reproduction of results is easy because all important information is stored in one place, namely the geometric model.

Name of Discusser: B. Simon, MTU Aero Engines Munich

Question:
You have stored your boundary condition on 3-D surfaces. Does that mean that boundary conditions are restricted on 3 types?

Answer?
No, each surface in the geometry has a list of virtual boundary conditions linked. This list can be as long as needed and each virtual boundary condition contains one type of information (e.g. pressure, temperature, heat transfer coeff.)